Lab Assignment 8: Markov Decision Processes and Dynamic Programming

Pawan Meena
Computer Science and Engineering
IIIT Vadodara
202351102@iiitvadodara.ac.in

Solanki Kuldipkumar Kishorbhai
Computer Science and Engineering
IIIT Vadodara
202351136@iiitvadodara.ac.in

Karan Haresh Lokchandani
Computer Science and Engineering
IIIT Vadodara
202351055@iiitvadodara.ac.in

***Abstract*—This report presents implementations and analysis of three fundamental Markov Decision Process (MDP) problems using Dynamic Programming algorithms. We implement Value Iteration for a stochastic Grid World environment and Policy Iteration for a bicycle rental inventory management problem. Our results demonstrate the effectiveness of these algorithms in finding optimal policies under uncertainty. The Grid World experiments explore how varying step costs influence optimal behavior, while the Gbike Rental problems illustrate realistic constraints such as free employee shuttles and parking limitations. All implementations and visualizations are available on GitHub[1].**

***Index Terms*—Markov Decision Process, Dynamic Programming, Value Iteration, Policy Iteration, Reinforcement Learning**

I. Introduction

Markov Decision Processes (MDPs) provide a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker. MDPs are fundamental to Reinforcement Learning (RL) and are widely used in robotics, operations research, and automated planning.

Dynamic Programming (DP) methods offer a powerful approach to solving MDPs when a complete model of the environment is available. Two key DP algorithms are:

- **Value Iteration**: Iteratively updates value estimates until convergence.
- **Policy Iteration**: Alternates between policy evaluation and improvement.

In this lab, we investigate three problems:

1) **Grid World**: A 4×3 stochastic navigation problem with varying step costs.
2) **Gbike Rental (Original)**: Inventory management with Poisson-distributed demands.
3) **Gbike Rental (Modified)**: Extended version with operational constraints.

II. Methodology

A. Grid World Value Iteration

The Grid World is a classic RL environment consisting of a 4×3 grid with:

- **Terminal States**: +1 reward at (0,3) and -1 reward at (1,3)
- **Wall**: Impassable obstacle at (1,1)
- **Stochastic Transitions**: 0.8 probability of intended direction, 0.1 each for perpendicular directions
- **Actions**: Up, Down, Left, Right

The Value Iteration update rule is:

$$V_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_k(s')] \tag{1}$$

We solve for five different step costs: $r(s) \in \{-0.04, -2, 0.1, 0.02, 1\}$ with discount factor $\gamma = 1.0$.

B. Gbike Bicycle Rental

This problem models bike-sharing inventory management across two locations:

- **States**: (n_1, n_2) where $n_i \in [0, 20]$ represents bikes at location i
- **Actions**: Net bikes moved from location 1 to 2, $a \in [-5, 5]$
- **Dynamics**: Poisson-distributed requests ($\lambda_1 = 3, \lambda_2 = 4$) and returns ($\lambda_1 = 3, \lambda_2 = 2$)
- **Rewards**: +INR 10 per rental, -INR 2 per bike moved
- **Discount**: $\gamma = 0.9$

We use Policy Iteration, alternating between:

1) **Policy Evaluation**: Compute V^π for current policy π
2) **Policy Improvement**: Update $\pi(s) = \arg\max_a Q^\pi(s,a)$

For efficiency, we precompute transition probabilities by exploiting the independence of locations.

C. Modified Gbike Problem

The modified version adds two realistic constraints:

1) **Free Shuttle**: An employee transports 1 bike from location 1 to 2 for free each night
2) **Parking Overflow**: INR 4 cost if >10 bikes are kept overnight at a location

These modifications affect the cost function:

$$C(s,a) = \begin{cases} (a-1)\times 2 & \text{if } a > 0 \\ |a| \times 2 & \text{if } a \leq 0 \end{cases} + 4\times\mathbb{I}[n_1 > 10] + 4\times\mathbb{I}[n_2 > 10] \tag{2}$$

[1] https://github.com/PhantomInTheWire/lab8

III. Results

A. Grid World Analysis

Fig. 1 shows the optimal policy for $r(s) = -0.04$, representing a small penalty per step. The agent takes the safest path to the +1 terminal state, avoiding the -1 state.

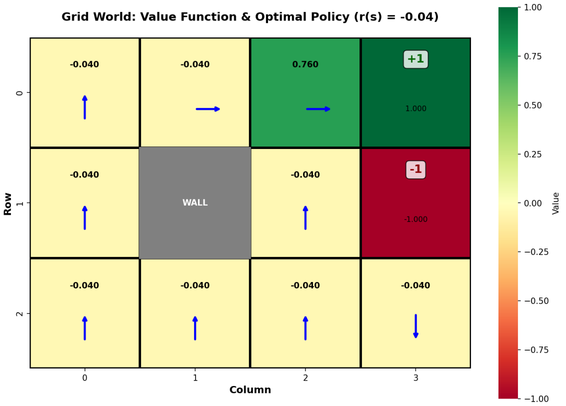


Fig. 1. Grid World with $r(s) = -0.04$: Balanced risk-averse policy.

Fig. 2 demonstrates extreme behavior with $r(s) = -2$. The high step cost makes the agent rush to the nearest terminal state, even if it's the -1 penalty state.

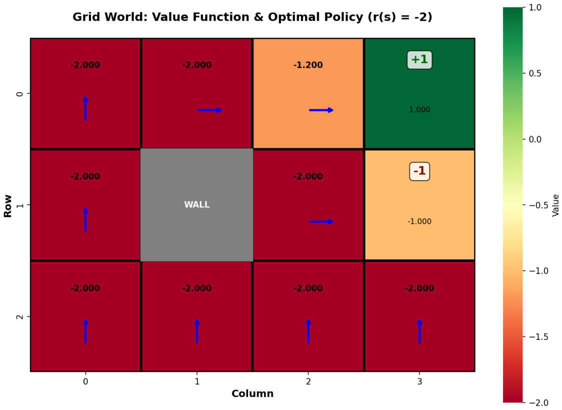


Fig. 2. Grid World with $r(s) = -2$: Rushes to nearest terminal.

Conversely, positive step costs ($r(s) = 0.1$ or 1) incentivize wandering behavior, as shown in Fig. 3, where the agent maximizes time before termination.

B. Gbike Rental Results

The original Gbike problem converged in 4 iterations. Fig. 4 shows the optimal policy heatmap, where positive values indicate transfers from location 1 to location 2.

The policy transfers bikes from the surplus location (1) to the deficit location (2), with the magnitude decreasing as location 2's inventory increases. Fig. 5 visualizes the value function, showing highest returns when both locations have balanced inventory.

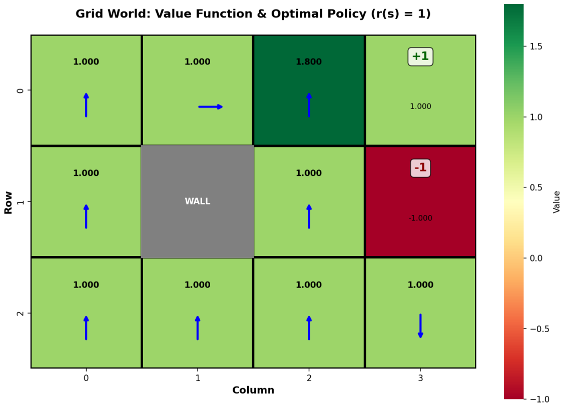
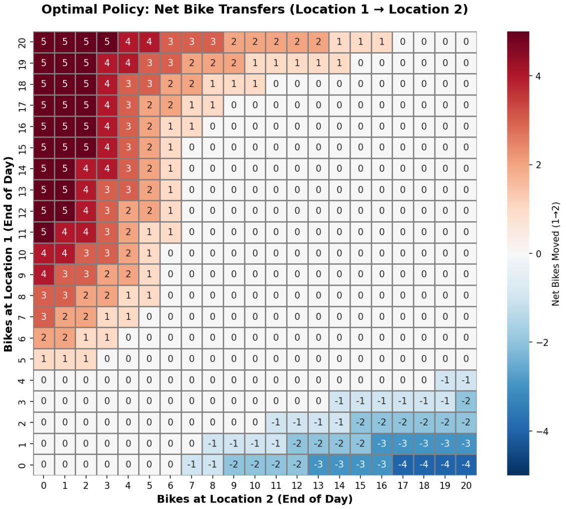


Fig. 3. Grid World with $r(s) = 1$: Wandering behavior to accumulate rewards.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
20	5	5	5	5	4	4	3	3	3	2	2	2	2	2	1	1	1	0	0	0	0
19	5	5	5	4	4	3	3	2	2	2	1	1	1	1	1	0	0	0	0	0	0
18	5	5	5	4	3	3	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0
17	5	5	5	4	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
16	5	5	5	4	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	5	5	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	5	5	4	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	5	5	4	3	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	5	5	4	3	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	5	4	4	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	4	4	3	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	4	3	3	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	3	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-2
2	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-2	-2
1	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-3	-3	-3	-3	-3
0	0	0	0	0	0	0	0	-1	-1	-2	-2	-2	-2	-3	-3	-3	-3	-4	-4	-4	-4

Fig. 4. Optimal Policy for Original Gbike Problem.

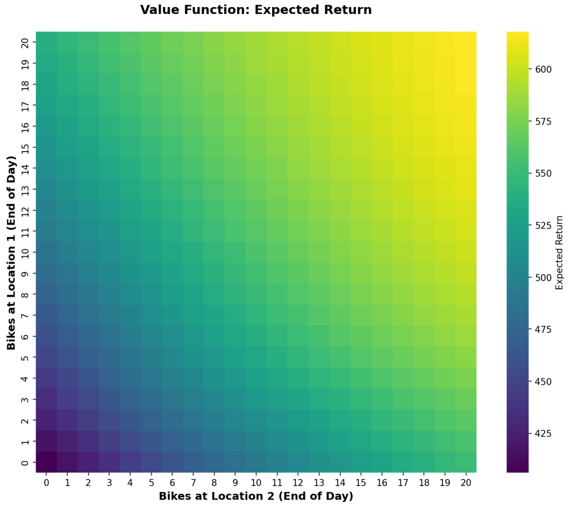


Fig. 5. Value Function Heatmap for Original Gbike Problem.

C. Modified Gbike Comparison

The modified problem converged in 3 iterations. Fig. 6 shows the adjusted policy accounting for the free shuttle and parking costs.

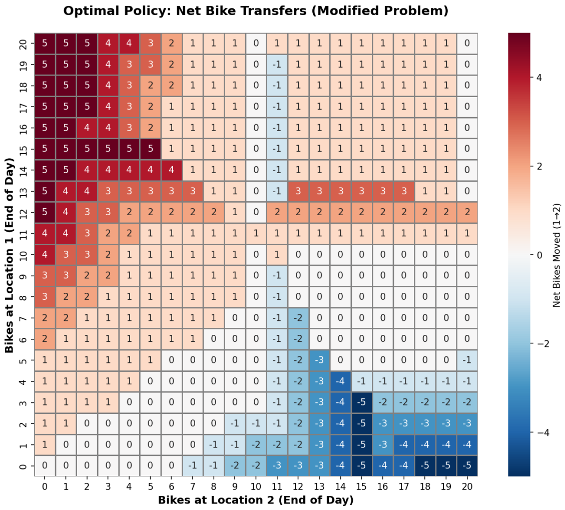


Fig. 6. Optimal Policy for Modified Gbike Problem.

Key observations:

- More frequent single-bike transfers (exploiting free shuttle)
- Avoidance of states with > 10 bikes to minimize parking costs
- Similar overall transfer strategy but with refined boundaries

IV. Conclusion

This lab demonstrated the effectiveness of Dynamic Programming for solving MDPs. Value Iteration successfully identified optimal policies across varying reward structures in the Grid World, illustrating how incentives shape agent behavior. Policy Iteration efficiently solved the complex Gbike Rental problem with Poisson dynamics, and the modified version showed how algorithmic solutions adapt to realistic operational constraints.

The implementations highlight the power of model-based RL when environment dynamics are known. Future work could explore model-free approaches for scenarios where transition probabilities are unknown.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.